

## MOTHERBOARD FOR SUPPORTING MULTIPLE GRAPHICS CARDS

### CROSS REFERENCE TO RELATED APPLICATIONS

**[00001]** Not Applicable.

### SPONSORED RESEARCH OR DEVELOPMENT

**[00002]** Not Applicable.

### SEQUENCE LISTINGS

**[00003]** Not Applicable.

### BACKGROUND OF THE INVENTION

#### Field of the Invention

**[00004]** The present invention provides a computer configured to effectuate the use of multiple, off-the-shelf video cards, working in parallel.

#### Discussion of the Related Art

**[00005]** Constant further improvements in graphic performance in computers are needed and desired by consumers. For instance, computers are increasingly used as digital entertainment hubs in the home to perform an array of demanding content creation and data manipulation tasks, including video editing and encoding, complex image processing, HDTV decoding, multichannel audio capture and playback, and of course far more realistic 3-D gaming. Furthermore, greater Internet bandwidth capabilities through the adoption of various high-speed access

technologies has resulted in the increased importance of graphics-based processing in online activities. For instance, online merchants provide increasing amounts of visual information to consumers who rely on the visual accuracy of the images in making purchasing decision. The list goes on, including applications like true voice recognition and synthesis, robust and accurate biometrics, and advanced encryption. High-end computers and workstations are also used by professionals for more computer-intensive scientific and engineering calculations, visualization and simulation, film-quality 3-D animation and rendering, advanced financial modeling, and numerous other heavy-duty chores.

**[00006]** Known methods for improving computer graphics performance are described below. In general, these improvements in computer graphics performance are achieved through developments in video card technology and enhancements in computer system architecture to maximize the gains in the video card performance.

#### Video Cards

**[00007]** Even before the beginning of the widespread use of personal computers, computer graphics has been one of the most promising and most challenging, aspects of computing. The first graphics personal computers developed for mass markets relied on the main computer processing unit ("CPU") to control every aspect of graphics output. Graphics boards, or video cards, in early systems acted as simple interfaces between the CPU and the display device and did not conduct any processing of their own. In other words, these early video cards simply

translated low level hardware commands issued by the CPU into analog signals which the display devices transformed into on-screen images. Because all of the processing was conducted by the CPU, graphics-intensive applications had a tendency to over-utilize processing cycles and prevent the CPU from performing other duties. This led to overall sluggishness and degraded system performance.

**[00008]** To offload the graphics workload from the CPU, hardware developers introduced video cards equipped with a Graphic Processing Unit (“GPU”). GPUs are capable of accepting high level graphics commands and processing them internally into the video signals required by display devices. By way of an extremely simplistic example, if an application requires a triangle to be drawn on the screen, rather than requiring the CPU to instruct the video card where to draw individual pixels on the screen (i.e., low level hardware commands), the application could simply send a “draw triangle” command to the video card, along with certain parameters (such the location of the triangle’s vertices), and the GPU could process such high level commands into a video signal. In this fashion, graphics processing previously performed by the CPU is now performed by the GPU. This innovation allows the CPU to handle non-graphics related duties more efficiently.

**[00009]** The primary drawback with early GPU-based video cards was that there was no set standard for the “language” of the various high level commands that the GPUs could interpret and then process. As a result, every application that sought to utilize the high level functions of a GPU based video card required a

specialized piece of software, commonly referred to as a driver, which could understand the GPU's language. With hundreds of different GPU-based video cards on the market, application developers became bogged down in writing these specialized drivers. In fact, it was not uncommon for a particularly popular software program to include hundreds, if not thousands, of video card drivers with its executable code. This, of course, greatly slowed the development and adoption of new software. This language problem was resolved by the adoption in modern computer operating systems by standardizing methods of video card interfacing. As a result, modern operating systems, such as the Windows® based operating system (sold by Microsoft Corporation of Redmond, WA), require only one hardware driver to be written for a video card. An intermediate software layer called an Application Programming Interface ("API") mediates interaction between the various software applications, the CPU and the video card. As a result, all that is required is that the video drivers and the applications be able to interpret a common graphics API. The two most common graphics APIs in use in today's personal computers are DirectX®, also distributed by Microsoft Corporation, and OpenGL®, distributed by a consortium of other computer hardware and software interests.

**[00010]** Since the advent of the GPU-based graphics processing subsystem, most efforts to increase the throughput of personal computer graphics subsystems (i.e., make the subsystem process information faster) have been geared, quite

naturally, toward producing more powerful and complex GPUs, and optimizing and increasing the capabilities of their corresponding APIs.

**[00011]** The graphics performance of a computer may also be improved through the use of multiple video cards, each with its own or multiple GPUs, processing graphics data in parallel. For example, co-pending and commonly assigned U.S. Patent Application No. 10/620,150 entitled MULTIPLE PARALLEL PROCESSOR COMPUTER GRAPHICS SYSTEM, the subject matter of which is hereby incorporated by reference in full, describes a scheme in which the display screen is divided into separate sections, and separate video cards are dedicated to the graphics processing in each of the display sections. It should be appreciated that numerous other technologies and methodologies for improving graphic performance schemes are also known, as described in the background section of U.S. Patent Application No. 10/620,150.

#### Improvements in Computer Architecture

**[00012]** A computer historically comprises a CPU that communicates to various other devices via a set of parallel conductors called a bus. When first introduced, computers only had one bus and were thus called single bus systems. As depicted in FIG. 1, a bus generally includes control lines, address lines and data lines that, combined, allow the CPU to oversee the performance of various operations (e.g., read or write) by the attached devices. Specifically, the CPU uses the control lines to control the operations of the attached devices and the address

lines to reference certain memory locations within the device. The data lines then provide an avenue for data transferred to or from a device.

**[00013]** Originally, most buses were set to run at a specified speed, measured in hertz or cycles per second. The CPU and the other various devices attached to the bus transferred data at different speeds, some faster than others. If the bus speed is unregulated, the different transfer speeds of the various components could potentially cause communications problems. Specifically, data transfer errors occur when relatively slower communicating components miss or lose messages from other components. To avoid this problem, the clock bus speed was set at a sufficiently slow speed so that all the components can communicate relatively error free through the bus.

**[00014]** This configuration, however, creates significant performance limitations, because data transfer rates are restricted to the levels of the slowest communicating components on the bus, thus preventing the relatively faster devices from realizing their full potential. The overall system performance could be improved by increasing the throughput (data transfer rates) for all of the devices on the bus and by similarly increasing the fixed bus speed. However, the system-wide improvement is relatively complex and expensive to implement.

**[00015]** To address the above-described problems, a multi-bus configuration may be used. In a multi-bus configuration, faster devices are placed on separate,

higher speed buses linked directly to the processor, thus allowing these high throughput devices to work more productively. For instance, it is common to have a separate local bus for graphics processors and other high throughput devices. This configuration thereby allows the high throughput devices to communicate without hindrance from the limitations of other devices.

**[00016]** There are several known ways to create a faster bus. As suggested above, increasing the speed of the bus (clock speed) allows more data transfers to take within a certain time. The capacity of the bus may also be achieved by increasing the width of the bus (i.e., increasing the amount of information being transferred on the bus at a particular instant). Referring back to FIG. 1, an increase in the number of address lines would effectively increase the number of addressable memory locations. Similarly, an increased number of data lines would enable more data bits to be sent at a time.

**[00017]** As described above, a computer may use various buses or a combination of buses. Currently known types of buses are summarized below in TABLE 1:

**TABLE 1**

<b>Bus Type</b>	<b>Max Clock Speed</b>	<b>Max Word Length</b>	<b>Comments</b>
Industrial Standard Architecture (ISA)	8 MHz	8 or 16 bits	<ul style="list-style-type: none"><li>• Requires two clock ticks for 16 bit data transfer</li><li>• Very slow for high performance disk accesses and high performance video cards</li></ul>
Enhanced Standard Architecture (EISA)	8.33 MHz	32-bit	<ul style="list-style-type: none"><li>• Can support lots of devices</li><li>• Supports older devices which have slower or smaller word lengths</li><li>• Transfers data every clock tick.</li></ul>
Micro channel Architecture (MCA)	10 MHz	32-bit	<ul style="list-style-type: none"><li>• Transfers data every clock tick.</li></ul>
Video Electronics Standard Association (VESA)/ Enhanced Video Electronics Standard Association Local Bus (VL)	33 MHz	32-bit	<ul style="list-style-type: none"><li>• Cannot take advantage of 64-bit architecture.</li><li>• Restricted on the number of devices, which can be connected (1 or 2 devices).</li></ul>
Peripherals Component Interconnect (PCI)	33 or 66 MHz	32 or 64 bit	<ul style="list-style-type: none"><li>• The PCI bus has a special chipset which allows more sophisticated control over the devices;</li><li>• PCI Bus can support many devices</li></ul>
Peripheral Component Interconnect Extended (PCI-X)	66 or 133 MHz	64 bit	<ul style="list-style-type: none"><li>• Primarily in computer servers</li></ul>

**[00018]** Currently, most personal computer systems rely on a PCI bus to connect their different hardware devices. PCI is a 64-bit bus, though it is usually



implemented as a 32-bit bus. A PCI bus runs at clock speeds of 33 or 66 MHz. At 32 bits and 33 MHz, the PCI local bus standard yields a throughput rate of 133 MBps. In the case of video cards, the bandwidth of the PCI bus has become increasingly limiting.

**[00019]**      Related to PCI, Peripheral Component Interconnect Extended (PCI-X) is a computer bus technology that increases the speed that data can move within a computer from 66 MHz to 133 MHz. Thus, PCI-X potentially doubles the speed and amount of data exchanged between the computer processor and peripherals. With PCI-X, one 64-bit bus runs at 133 MHz with the rest running at 66 MHz, allowing for a data exchange of 1.06 GB per second. PCI-X, however, is used primarily in computer servers, and not in desktop computers.

**[00020]**      In response to the bandwidth limitations of the PCI Bus, the Accelerated Graphics Port ("AGP") bus was developed for use with graphics processing devices, and most high performance video cards currently connect to the computer exclusively through a dedicated AGP slot found on the motherboard. AGP is based on PCI but is designed especially for the throughput demands of 3-D graphics. Rather than using the PCI bus for graphics data, AGP introduces a dedicated point-to-point channel so that the graphics controller can directly access main memory. The AGP channel is 32 bits wide and runs at 66 MHz. This translates into a total bandwidth of 266 MBps, as opposed to the PCI bandwidth of 133 MBps. AGP also supports three optional faster modes, with throughputs of 533

MBps (2x), 1.07 GBps (4x), and 2.14 GBps (8x). In addition, AGP further improves graphics performance by allowing graphics-related data and 3-D textures to be stored in main memory rather than video memory.

**[00021]** As the major hardware subsystems get faster, at different rates and move more data around, PCI and other currently used interconnects just cannot handle the load. Also, with the increasingly powerful and complex GPUs and better optimized and capable APIs, bus bandwidth limitations are again becoming a primary limitation to graphic system performance. Furthermore, many current and emerging tasks need faster processors, graphics, networking, and storage subsystems, and that translates into a need for much faster interconnects between those subsystems. Accordingly, new types of scalable bus standards, such as PCI Express (described in greater detail below), are being developed to address these limitations while preserving compatibility with existing components.

**[00022]** Despite the above-described innovations and other known advances for enabling improvements in computer graphic performance, there remains a continuous need for further improvements. For commercial viability, these improvements should use commonly available, off-the-shelf components. Furthermore, the improvements should not require extensive changes in hardware or software, so that the improved computer retains general compatibility with existing components and applications.

**[00023]** No known, commonly available computer currently uses two or more high performance graphics cards.

#### BRIEF SUMMARY OF THE PRESENT INVENTION

**[00024]** In response to these and other needs, the current invention provides a system and method for supporting two or more high bandwidth PCI Express graphics slots on a single motherboard, each capable of supporting a commonly available, off-the-shelf video card. In one embodiment, the motherboard chipset supports at least 32 PCI Express lanes, with these lanes being routed into two x16 PCI Express graphics slots. In another embodiment, the motherboard chipset supports at least 24 PCI Express lanes, with 16 lanes being routed into one x16 PCI Express graphics slot, and the remaining eight lanes being routed into one x8 PCI Express graphics slot (which slot physically could use the same connector used by the x16 PCI Express graphics slot, but it would only have eight PCI Express lanes “active”). In yet another implementation, the present invention splits the 16 lanes dedicated to the x16 connect, enabling two x8 PCI Express graphics slots (which slots physically could use the same connector used by the x16 PCI Express graphics slot, but would only have eight PCI Express lanes “active”). And finally, the present invention can use a PCI Express switch that converts the 16 lanes coming from the chipset “root complex” into two x16 links that connect two x16 PCI Express graphics slots. Importantly, each and every embodiment of the present invention is agnostic to a specific chipset (e.g., Intel, AMD, etc.).

## BRIEF DESCRIPTION OF THE DRAWINGS

**[00025]** These and other advantages of the present invention are described more fully in the following drawings and accompanying text in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a schematic, high-level illustration of a conventional computer bus;

FIGS. 2A-2B are schematic illustrations of conventional PCI Express motherboards;

FIGS. 3A-3B, 4 and 5A-5B are schematic illustrations of the operations of PCI Express motherboards; and

FIGS. 6-10 depict schematic illustrations of a PCI Express motherboard containing multiple video cards in accordance with the various embodiments of the present invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[00026]** The present invention exploits the below-described PCI Express interconnect to provide a motherboard that supports two or more high bandwidth PCI Express graphics slots, each capable of supporting a commonly available, off-the-shelf video card.

**[00027]** PCI Express®, as depicted in FIGS. 2A and 2B (Prior Art), is a new type of computer interconnect that will supplant the widely used PCI, PCI-X and AGP buses described above. PCI Express is a high-performance interconnect that

gives more for less, meaning more bandwidth with fewer pins. PCI Express is designed to leverage the strengths of yesterday's and current general I/O architectures while addressing immediate and future I/O architectural and mechanical issues with current technologies. A few examples of these issues are bandwidth constraints, protocol limitations and high pin count. More technically speaking, PCI Express is a high speed, low voltage, differential serial pathway for two devices to communicate with each other. PCI Express uses a protocol that allows devices to communicate simultaneously by implementing dual unidirectional paths between two devices.

**[00028]** Compared to the shared, parallel bus architecture of PCI and past buses, point-to-point connections permit each device to have a dedicated link without arbitrating for a shared bus. PCI Express is targeted at chip-to-chip I/O interconnects, expansion card connections, and it can act as an I/O attach point on the motherboard for other interconnects such as USB 2.0, InfiniBand, Ethernet, and 1394/1394b.

**[00029]** A quick overview of a PCI Express-based motherboard is now provided. The following description of PCI Express is meant for illustrative purposes and is not intended to limit the present invention. The PCI Express interconnect is still under development and refinement. It is anticipated that PCI Express will evolve and change as needed, and thus, these changes should fall within the present invention. A complete understanding of PCI Express is generally outside the scope

of the current application, and more information on PCI Express can be found at [www.express-lane.org](http://www.express-lane.org). For additional information on PCI Express, please also refer to Don Anderson, et al., PCI EXPRESS SYSTEM ARCHITECTURE; Adam Wilen, et al., INTRODUCTION TO PCI EXPRESS: A HARDWARE AND SOFTWARE DEVELOPER'S GUIDE; and Ed Solari And Brad Congdon, COMPLETE PCI EXPRESS REFERENCE, THE: DESIGN INSIGHTS FOR HARDWARE AND SOFTWARE DEVELOPERS. The subject matter of these three books is hereby incorporated by reference in full.

**[00030]** Returning now to FIGS. 2A and 2B, the motherboard 200 or 201 has one or more CPUs 210 connected to various components via a chipset 220. The CPU 210 is the brain of the computer, where most calculations take place. In modern computers, the CPU 210 is housed in a single chip called a microprocessor. Two typical components of the CPU 210 are an arithmetic logic unit ("ALU"), which performs arithmetic and logical operations and a control unit that extracts instructions from memory and decodes and executes them, calling on the ALU when necessary.

**[00031]** Continuing with FIGS. 2A and 2B, the CPU 210 is generally connected to a VRM 211 and a clock 212. The VRM 211, short for voltage regulator module, regulates the power supplied to the CPU 210, typically in the range 3.3V. The VRM may also carry out a dual voltage or "split rail" voltage scheme in which the CPU 210 receives an external or I/O voltage, typically 3.3V, and a lower internal or core

voltage, usually 2.8V to 3.2V. The clock 212 is an oscillator, typically a quartz-crystal circuit similar to those used in radio communications equipment, that sets the tempo for the processor. Clock speed is usually measured in MHz (megahertz, or millions of pulses per second) or GHz (gigahertz, or billions of pulses per second).

**[00032]** A chipset 220 is a group of microchips designed to work and sold as a unit to connect the CPU 210 to the various components on the motherboard 200, 201. It should be appreciated that the various motherboards described in this application, the various combination components connected to the motherboard, and the chipset 220 may be adapted as needed to meet commercial and practical needs. Thus, the following description of the motherboard 200, 201 and the implementation of the chipset 220 is provided merely for illustration and should not be used to limit the present invention. Importantly, each and every embodiment of the present invention is agnostic to a specific chipset (e.g., Intel, AMD, etc.) or the scalable, high-speed bus employed by the chipset.

**[00033]** In one current implementation of a PCI Express motherboard, the chipset 220 includes a memory controller hub (MCH) 221 and an input/output (I/O) bridge 222. The MCH 221 is a host bridge that provides a high-speed, direct connection from the CPU 210 to memory 240 and a video or graphics card 270. The PCI Express connection 271 between the MCH 221 and the graphics card 270 is described in greater detail below. Similar to the MCH 221, the I/O bridge 222 regulates connections between the CPU 210 and the other components on the

motherboard 200, 201. The MCH 221 and the I/O bridge 222 are relatively well-known microchips. For instance, Intel® Corp. of Santa Clara, California produces an 875 chipset that includes an 82875 MCH microchip and an 82801EB or 82801ER I/O Controller Hub (ICH) microchip. While a PCI Express chipset will differ, the PCI Express chipset may use similar components.

**[00034]** The various components connecting to the CPU 210 via the I/O bridge 222 are now summarized. For instance, the I/O bridge 222 may connect the CPU 210 to various I/O connections 230. These I/O connections 230 include universal serial bus (USB) 231, Local I/O 232, and disk connections 233 such as Serial Advanced Technology Attachment (SATA).

**[00035]** USB 231 is a plug-and-play interface between a computer and add-on external devices (such as audio players, joysticks, keyboards, telephones, scanners, printers, etc.). With USB connections 231, a new device can be added to a computer without having to add an adapter card or even having to turn the computer off.

**[00036]** The Local I/O connection 232, such as a low pin count (LPC) interface, connects the CPU 210 to various components on the motherboard. The LPC Interface allows the legacy I/O motherboard components, typically integrated in a Super I/O chip, to migrate from the ISA/X-bus to the LPC interface, while retaining full software compatibility. The LPC Specification offers several key advantages over ISA/X-bus, such as reduced pin count for easier, more cost-effective design.



The LPC interface is software transparent for I/O functions and compatible with existing peripheral devices and applications and describes memory, I/O and DMA transactions. Unlike ISA, which runs at 8 MHz, the LPC interface uses the PCI 33 MHz clock. LPC memory consists of a flash memory with an LPC interface built in and is designed to replace standard flash for storing the BIOS on PC motherboards. Communicating over the LPC bus allows larger memory with fewer pins.

**[00037]** As described above, the local I/O connection 232 may include a single Super I/O chip that, much like the system chipset, performs many functions that used to take several pieces of hardware in the past. This design standardizes and simplifies the motherboard and, thus, reduces cost. The Super I/O chip typically is responsible for controlling the slower-speed, mundane peripherals found in every computer. Since these devices have been mostly standardized, they are virtually the same on every PC, and it is easier to integrate these into a commodity chip instead of worrying about them for each motherboard design. The major functions of the Super I/O controller chip are Serial Port Control, Parallel Port Control and Floppy Disk Drive Control. A Super I/O controller chip may further integrate other functions as well, such as the real-time clock, keyboard controller, and, in some cases, even the IDE hard disk controllers.

**[00038]** The hard drive connection 233 connects mass storage devices (e.g., hard disk or CD-ROM drive) to computer systems. As its name implies, SATA is based on serial signaling technology, unlike Integrated Drive Electronics (IDE) hard

drive connections that use parallel signaling. With either SATA or IDE interfaces, a controller is integrated into the mass storage devices. Either types of interfaces may support Ultra Direct Memory Access (DMA), a protocol the enables burst mode data transfer rates of 33.3 MBps, and ATA/100, an updated version of ATA that increases data transfer rates to 100 MBps. (triple the standard of 33 MBps).

**[00039]** Continuing with FIGS. 2A-2B, the I/O Bridge 222 (such as the proposed Intel ® 41210 Serial-to-Parallel PCI Bridge) may include a PCI Express-to-PCI bridge that enables existing PCI/PCI-X adapters and add-in cards to connect to the motherboard 200, 201 via the PCI connections 250.

**[00040]** The various PCI-Express adapters and add-in cards may connect to the motherboard 200, 201 through the PCI Express connections 260. The details of the PCI Express connections 260 are described in greater detail below in the following discussion of the operations of the PCI Express bus. As depicted in FIG. 2B, the PCI Express motherboard 201 may include a switch 280 that distributes data between the I/O bridge 222 and various components connected to the PCI connections 260.

**[00041]** A PCI Express connection, or link, is based on lanes. A lane is a serial link capable of establishing a bi-directional communication between two hardware devices ("end points"). A single, basic PCI Express serial link (as described below in FIG. 5 and the accompanying text) is a dual-simplex connection using two low-

voltage pairs of differentially driven signals -- a receive pair and a transmit pair (four wires). A differential signal is derived by using the voltage difference between two conductors. The first-generation PCI Express link signaling speed is 2.5Gbits/sec per wire pair (in each direction), and a 5Gbit/sec link may become available by the time PCI Express ships in volume in early 2004.

**[00042]** A dual simplex connection permits data to be transferred in both directions simultaneously, similar to full duplex connections (as in telephones), but with dual simplex, each wire pair has its own ground unlike full duplex, which uses a common ground. Higher speed and better signal quality is attainable with dual simplex connections. With the PCI bus, for instance, an initiating device must first request access to the shared PCI bus from a central arbiter, and then take control of the bus to transfer data to a target device, with data transfers occurring in one direction between two devices at any given point in time.

**[00043]** Another key feature of the basic PCI Express serial link is its embedded clocking technique using 8b/10b encoding. The clock information is encoded directly into the data stream, rather than having the clock as a separate signal. As described below, the 8b/10b encoding essentially requires 10 bits per 8 bit character, or about 20% channel overhead.

**[00044]** The PCI Express connections 260, 271 may be comprised of multiple lanes. Each lane is comprised of the two differentially driven pair of wires

(transmit and receive) of a basic link, as mentioned earlier. The lanes may scale 2.5Gbit/sec in each direction to 10Gbits/sec and beyond in the future. Multiple lanes can be connected between devices, chips, etc. While operating similarly to parallel interfaces, each of the lanes is actually a pair of grouped independent serial connections, thus avoiding the signal quality problems cited earlier for parallel interfaces.

**[00045]** A PCI Express link can have single lane (x1) or multiple lanes can be combined (e.g., x2, x4, x8, x12, x16, and x32 lane widths). For example, combining two lanes produces a x2 link (read “by” 2), combining four lanes produces a x4 link, and so forth (x8, x16, x32). For most applications, a x1 link (i.e., single lane) will suffice. Given a x1 link has 4 wires (two differential signal pairs, one in each direction), a x16 link would have sixteen differential signal pairs in each direction, or sixty-four wires for bi-directional data. At the high end, a x32 link can transmit 10GB/sec each direction (2.5Gbits/sec x 32 / 8 bits). But with 8b/10b encoding, the transmission rate is actually in the range of 8GB/sec because of the 20% embedded clock overhead.

**[00046]** The links in PCI Express are symmetric and cannot be configured asymmetrically, with more lanes in one direction versus the other. Furthermore, lane ordering can be swapped per device, and polarities of the positive and negative conductors of a differential signal pair can be inverted at the receiver to provide design flexibility and help avoid physical signal crossovers in layout.

**[00047]** As mentioned above, PCI Express uses a packetized and layered protocol structure, and it does not require any sideband signaling riding alongside the main serial interconnection as sometimes used in AGP. Layered protocols have been used for years in data communications and permit isolation between different functional areas in the protocol, and allow updating/upgrading different layers often without requiring changes in the other layers. For example, new transaction types might be included in newer revisions of a protocol specification that does not affect lower layers, or the physical media might be changed with no major effects on higher layers.

**[00048]** Graphics cards will generally need more than a x1 link. In fact, due to the high amount of data that needs to be transferred to a graphics card, it has been established that all currently planned PCI Express motherboards will have a single x16 PCI Express slot dedicated solely to support a graphics card. Thus, if a specific chipset supports twenty PCI Express lanes, sixteen lanes would be dedicated to the x16 graphics slot, and the remaining four lanes would be used for four x1 slots. All currently planned PCI Express motherboards only support a single x16 PCI Express graphics slot.

**[00049]** The PCI Express architecture is based on layers, as depicted in the PCI Express layer diagram 300 in FIG. 3A. Compatibility with the current PCI addressing model, a load-store architecture with a flat address space is maintained in PCI Express to ensure that all existing applications and drivers operate

unchanged. PCI Express configuration also generally uses standard mechanisms as defined in the PCI Plug-and-Play specification. The software layers in the PCI Express layer diagram 300 generates read and write requests that are transported by the transaction layer to the I/O devices using a packet-based, split-transaction protocol. The link layer adds sequence numbers and cyclic redundancy code (CRC) to these packets to create a highly reliable data transfer mechanism. CRC is in checking for errors in data transmissions on a communications link. A sending device applies a 16- or 32-bit polynomial to a block of data that is to be transmitted and appends the resulting CRC to the block. The receiving end applies the same polynomial to the data and compares its result with the result appended by the sender. If they agree, the data has been received successfully. If not, the sender can be notified to resend the block of data. Continuing with the PCI Express layer diagram 300, the basic physical layer (described in greater detail below) consists of a dual-simplex channel that is implemented as a transmit pair and a receive pair.

**[00050]** Referring now to FIG. 3B, a PCI Express data layer diagram 310 illustrates the relationship of the data between the different layers. As suggested above, the primary role of the link layer is to ensure reliable delivery of the packet across the PCI Express link. The link layer is responsible for data integrity and adds a sequence number and the CRC to the transaction layer packet, as shown in FIG. 3B. Most packets are initiated at the Transaction Layer using a credit-based, flow control protocol that ensures that packets are only transmitted when it is

known that a buffer is available to receive this packet at the other end. This configuration eliminates any packet retries, and their associated waste of bus bandwidth due to resource constraints. The Link Layer then automatically resends a packet that was signaled as corrupted.

**[00051]** Continuing with the PCI Express data layer diagram 310 of FIG. 3B, the transaction layer receives read and write requests from the software layer and creates request packets for transmission to the link layer. All requests are implemented as split transactions, and some of the request packets will need a response packet. The transaction layer also receives response packets from the link layer and matches these with the original software requests. Each packet has a unique identifier that enables response packets to be directed to the correct originator. The packet format supports 32-bit memory addressing and extended 64-bit memory addressing. Packets also have attributes such as “no-snoop”, “relaxed-ordering” and “priority” which may be used to optimally route these packets through the I/O subsystem.

**[00052]** The transaction layer in PCI Express is designed to support four address spaces, including three PCI address spaces (memory, I/O and configuration) and a Message Space. PCI 2.2 introduced an alternate method of propagating system interrupts called Message Signaled Interrupt (MSI). In MSI, a special-format memory write transaction was used instead of a hard-wired sideband signal. The PCI Express specification also uses the MSI concept as a primary method for

interrupt processing and uses Message Space to support all prior side-band signals, such as interrupts, power-management requests, resets, and so on, as in-band messages. Other special cycles within the PCI 2.2 specification, such as Interrupt Acknowledge, are also implemented as in-band messages. The PCI Express Messages function as virtual wires since they effectively eliminate the wide array of sideband signals currently used in a platform implementation.

**[00053]** Referring now to FIG. 4, a fundamental PCI Express x1 link 200 consists of two, low-voltage, differentially driven pairs of signals, a transmit pair and a receive pair. A data clock is embedded using the 8b/10b encoding scheme to achieve very high data rates, initially 0.25 Giga transfers/second/direction. Thus, the physical layer actually transports packets between the link layers of two PCI Express agents.

**[00054]** The transportation of byte data is depicted in a x1 lane byte data diagram 500 of FIG. 5A. Specifically, different packets of data are sent serially (i.e., one after the other) across the single lane. Each byte is transmitted across the lane with 8b/10b encoding, as described above.

**[00055]** As previously described, the bandwidth of a PCI Express link may be linearly scaled by adding signal pairs to form multiple lanes. The physical layer supports x1, x2, x4, x8, x12, x16 and x32 line (or greater) widths. With multiple lanes, the PCI Express interconnect splits the byte data as suggested in a multiple



lanes byte data diagram 510 of FIG. 5B. The multiple lanes byte data diagram 510 demonstrates the splitting of byte data for transport using x4 (four lane) connection. Specifically, the data is disassembled for parallel transport across the four lanes and then reassembled at the receiving end. This data disassembly and re-assembly is transparent to other layers.

**[00056]** During initialization, each PCI Express link is set up following a negotiation of lane widths and frequency of operation by the two agents at each end of the link. No firmware or operating system software is involved. The PCI Express architecture comprehends future performance enhancements via speed upgrades and advanced encoding techniques. The future speeds, encoding techniques or media would only impact the physical layer.

#### PCI Express Motherboard for Multiple Graphics Cards

**[00057]** By exploiting the above-described PCI Express interconnect, the present invention provides a motherboard that supports two or more high bandwidth PCI Express graphics slots, each capable of supporting a commonly available, off-the-shelf video card. Specifically, the current invention provides a system and method for supporting two or more high bandwidth (e.g., 2 x8 or higher bandwidth connections) PCI Express Graphics slots on a single motherboard. The integration of the two or more high bandwidth graphics cards may be accomplished in several ways.

**[00058]** Turning to FIG. 6, one embodiment of the present invention provides Multi-Video Card PCI Express Motherboard 600 that supports at least thirty-two PCI Express lanes, where these lanes are routed into two or more x16 PCI Express Graphics slots. Specifically, the Multi-Video Card PCI Express Motherboard 600 connects two or more graphics card 670 to the MCH 221, each connected via a 16x PCI Express connection 671. The performance of the multiple video cards may be synchronized using various known techniques. For instance, the above-referenced U.S. Patent Application No. 10/620,150 provides a scheme for coordinating the operations of multiple GPUs. In the present invention, the various GPUs are located on separate graphics cards, each connected to a high bandwidth PCI Express graphics slot.

**[00059]** In another implementation of the present invention depicted in FIG. 7, a Multi-Video Card PCI Express Motherboard 700 divides the sixteen lanes dedicated to the x16 connect to form a pair of x8 connections 771 for connecting the graphics cards 770 to the MCH 221. Specifically, the Multi-Video Card PCI Express Motherboard 700 may have two x8 graphics slots. In the same way, a x32 connection (or the pair of x16 connections depicted in FIG. 6) may be divided to form four x8 connections. While each of the x8 slots has a reduced bandwidth capacity in comparison to x16 slot, the capacity of the x8 still exceeds the current capabilities of many video cards. Furthermore, the performance of a pair of video cards 770

connected to the x8 slots will generally exceed the performance of a single video card connected to a x16 slot.

**[00060]** Referring now to FIG. 8, another embodiment of the present invention provides a Multi-Video Card PCI Express Motherboard 800 that connects two or more video cards 870 using a PCI Express switch 880. The PCI Express switch 880 converts the sixteen lanes 871 coming from the chipset 220 root complex into two or more distributed x16 links 872, each connected to a x16 PCI Express Graphics slot. When connected by the switch 880 to the chipset 220, a video card 870 may send a very large burst of data via the PCI Express connection 871 and the distributed x16 link 872. Because the video card 870 does not continuously export data at the capacity of the PCI Express connection 871, the use of the switch 880 better allows the Multi-Video Card PCI Express Motherboard 800 to exploit the large capacity of the x16 connection to the chipset 220.

**[00061]** In another implementation of the present invention depicted in FIG. 9, a Multi-Video Card PCI Express Motherboard 900 divides twenty-four lanes from the chipset 220 to form a x16 connection 971 and a x8 connection 972 to the graphics cards 970a and 970b, respectively. The graphics card slot associated with the x8 connection 972 is generally physically identical to the graphics card slot associated with the x16 connection 971. However, the x8 connection 972 provides approximately half the bandwidth. Thus, the graphics cards 970a and 970b may be substantially similarly and are generally interchangeable so long as the graphics

cards 970a and 970b detect the nature of the PCI Express connection (i.e., whether the connection is x8 or x16) and operate accordingly.

**[00062]** The various embodiments of the present invention may also be implemented using a Multi-Video Card PCI Express Motherboard 1000, as depicted in FIG. 10. In particular, the Multi-Video Card PCI Express Motherboard 1000 uses a splitter 1080 which routes data transfer from a PCI Express connection 1071 to multiple graphics cards 1070 via connections 1072 and 1073. In contrast to the switch 880 which allocates access to the chipset 220, the splitter 1080 merely physically divides lanes in the PCI Express connection 1071. For instance, a x16 connection may be divided into two x8 connections, a x 24 connection may be divided into x8 and x16 connections, a x32 connection may be divided into two x16 connections, and so on.

**[00063]** The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. In particular, it is foreseeable that different components using different data transfer standards may be added to a PCI Express motherboard. Furthermore, the present invention, while primarily describing a PCI Express motherboard having two high-speed video card slots, may

be adapted to provide any number of video card slots using the techniques described herein. The teachings of the present invention may also be combined to form various combinations of high-speed video slots. For instance, one of the 8x PCI Express connection 771 may be connected to a switch 880 to distribute the bandwidth of that 8x PCI Express connection 771 to two or more x8 video cards slots. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.